ELBUG
SUPPLEMENT TO BEEBUG

December 1985 – Volume 3 Issue 2

MAKING THE MOST OF THE FUNCTION KEYS (Part 1)

The simple user-definable function keys of the Electron offer a wealth of possibilities as Dave Robinson discovers.

To get the maximum benefit from any feature of the Electron it is necessary to try and understand how it is implemented in the computer's memory and how the feature is utilized by the Operating System.

The user-programmable function keys are designed to let you insert a series of characters, or instructions, into the keyboard buffer with only a 'single' keystroke. To use, press and hold the Func key down and then press the appropriate number key before releasing Func.

Pressing a function key (that has already been defined) does the same job as typing directly at the keyboard.

For example, try typing:

```
*KEYØ PRINT CHR$( <Return>
```

This will cause the characters PRINT CHR$(to appear on the screen, and in the keyboard buffer, every time the FUNC key and the number Ø key are pressed together. This would save you typing this each time it appeared in a long program listing you are entering.

Used in this way, several keys can be defined to help speed up your typing.

MINI-PROGRAMS

However, the real power of the function keys is their ability to store a mini-program. This mini-program is run every time that key is pressed. Try:

```
*KEY1 MODE6|M|NLIST|M <Return>
```

→

G CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE E

The December ELBUG magazine cassette is packed with programs for Electron owners. Many of the programs from this month's BEEBUG are there - the personal diary manager, the Chinese Chequers game, the Workshop procedures and demonstration program, the updated BEEBUG filer program, the First Course colour mixing program, and the full screen Basic editor - plus the function key buffer inspection program from this ELBUG supplement. All this for just £3.00 plus 50p post and packing.

This is a very useful definition that will program key 1 to cause the Electron to switch to mode 6, select paging mode (to stop the screen scrolling) and list any Basic program in memory.

CONTROL CODES

The secret of these mini-programs lies in the ability to put control codes into the key definitions.

These control codes are the ASCII numbers below &20 (32 in decimal). They are the codes that you would usually issue with the VDU statement. You will find a complete list of them on page 105 of the User Guide. The third column of the table on that page is headed Ctrl. In this column you will see a symbol against each control code. The key corresponding to this symbol when pressed along with the Ctrl key will produce the same effect as the control code (if issued as VDU or PRINT CHR$ statement).

From that table we see that Ctrl-N selects paging mode and that Ctrl-M has the same effect as the Return key.

Control codes are included in key definitions using the | character - equivalent to the Ctrl key - obtained by pressing Shift and the right cursor key together.

Quite long mini-programs can be defined using the Ctrl (|) codes to produce Returns and other effects (though, as we shall see later, there is a limit to their length).

```
*KEY2 C$="LIST"+STR$ ERL+CHR$13:A%=138
:X%=0:FOR L=1 TO LENC$:Y%=ASC(MID$(C$,L):C
ALL&FFF4:NEXT|M
```

This will define key 2 to list the line where the last error occurred.

In this example, colons have been used to separate instructions, just as they would be used in a Basic program.

Instead of colons to separate different statements, it is quite permissible to use line numbers in a function key definition and then include the keyword RUN before the final |M. Remember that you can perform any action from a function key definition that would normally be performed from the keyboard. However, this would mean that pressing the function key would create a program in the section of memory allocated for Basic programs and may overwrite an existing program. So it is better on the whole not to use program line numbers in the definition.

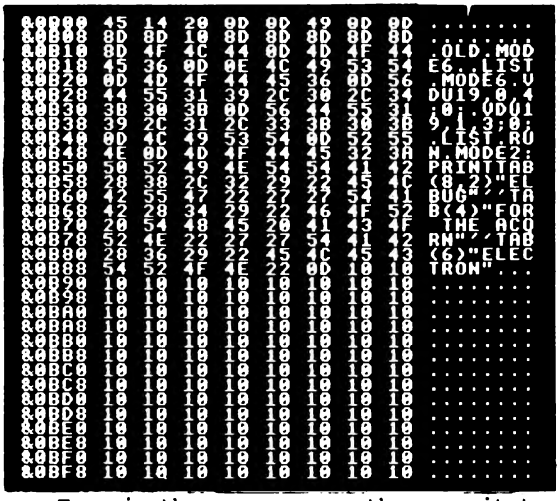
FUNCTION KEY BUFFER.

The Electron has to store the function key definitions somewhere in memory. The area used is called the Function Key buffer. It resides in page &B in RAM memory. This includes all addresses from &0B00 to &0BFF, a total of &100 (256) bytes.

Now 256 bytes may, or may not, seem a lot of room. All the definitions given to the function keys have to share this buffer space. In fact the actual number of bytes is reduced to 239 to allow the Operating System to keep track of which key definitions are where inside the buffer.

The following short program allows you to examine, in detail, how this buffer is arranged.

```
  10 REM PROGRAM f-KEY BUFFER
  20 REM VERSION E0.1
  30 REM AUTHOR  DAVE ROBINSON
  40 REM ELBUG   DEC 1985
  50 REM PROGRAM SUBJECT TO COPYRIGHT
  60 :
 100 MODE 4
 110 ON ERROR GOTO 270
 120 FOR address%=&B00 TO &BFF STEP 8
 130 PRINT '"&0";~address%;SPC1;
 140 B$=""
 150 FOR byte%=0 TO 7
 160 char%=address%?byte%
 170 char$=STRING$(-(char%<16),"0")+STR
$~(char%)
 180 PRINT char$;SPC1;
 190 IF char%<32 OR address%+byte%<&B11
 char%=46
 200 B$=B$+CHR$(char%)
 210 NEXT byte%
 220 PRINT B$;
 230 NEXT address%
 240 @%=&90A
 250 END
 260 :
 270 ON ERROR OFF
 280 @%=&90A
 290 IF ERR<>17 REPORT:PRINT " at line
";ERL
 300 END
```



```
&0B00 45 14 20 0D 0D 49 0D 0D .....
&0B08 8D 8D 10 8D 8D 8D 8D 8D .....
&0B10 8D 4F 4C 44 0D 4D 4F 44 .OLD.MOD
&0B18 45 36 0D 0E 4C 49 53 54 E6..LIST
&0B20 0D 4D 4F 44 45 36 0D 56 .MODE6.V
&0B28 44 55 31 39 2C 30 2C 34 DU19,0,4
&0B30 3B 30 3B 0D 56 44 55 31 ;0;.VDU1
&0B38 39 2C 31 2C 33 3B 30 3B 9,1,3;0;
&0B40 0D 4C 49 53 54 0D 52 55 .LIST.RU
&0B48 4E 0D 4D 4F 44 45 32 3A N.MODE2:
&0B50 50 52 49 4E 54 54 41 42 PRINTTAB
&0B58 28 38 2C 32 29 22 45 4C (8,2)"EL
&0B60 42 55 47 22 27 27 54 41 BUG"''TA
&0B68 42 28 34 29 22 46 4F 52 B(4)"FOR
&0B70 20 54 48 45 20 41 43 4F  THE ACO
&0B78 52 4E 22 27 27 54 41 42 RN"''TAB
&0B80 28 36 29 22 45 4C 45 43 (6)"ELEC
&0B88 54 52 4F 4E 22 0D 10 10 TRON"...
&0B90 10 10 10 10 10 10 10 10 ........
&0B98 10 10 10 10 10 10 10 10 ........
&0BA0 10 10 10 10 10 10 10 10 ........
&0BA8 10 10 10 10 10 10 10 10 ........
&0BB0 10 10 10 10 10 10 10 10 ........
&0BB8 10 10 10 10 10 10 10 10 ........
&0BC0 10 10 10 10 10 10 10 10 ........
&0BC8 10 10 10 10 10 10 10 10 ........
&0BD0 10 10 10 10 10 10 10 10 ........
&0BD8 10 10 10 10 10 10 10 10 ........
&0BE0 10 10 10 10 10 10 10 10 ........
&0BE8 10 10 10 10 10 10 10 10 ........
&0BF0 10 10 10 10 10 10 10 10 ........
&0BF8 10 10 10 10 10 10 10 10 ........
```

Type in the program and then use it to study the function key buffer before and after you enter some function key definitions.

Addresses &0B00 to &0B0F record, in order, the address offset (from &0B00) of the byte before the first character of all sixteen function keys definitions. Yes, sixteen! This includes Break - key 10 - and the four cursor keys and Copy key which can act as function keys 11 to 15 after a *FX4,2 command has been issued.

Address &0B10 keeps track of the last used address in the buffer. So when no keys have been defined - when you first switch on the computer, for example, or after a hard reset (Ctrl-Break) - this location contains its own address in the buffer (&10) as it is the last used byte. (In fact, in this situation every byte in the buffer will contain the byte, &10). Conversely, if &0B10 contains &FF then the buffer is full.

The rest of the buffer - from &0B11 to &0BFF contains the ASCII codes of the definitions as typed in. Unlike Basic programs in main memory, keywords in the function key buffer are not tokenized (reduced to a single byte) but are stored as they are typed in.

Any control codes will be stored as their ASCII value, using only one byte and not the two that you might expect. For example |M will be stored as &0D (13).

Note that when a new definition for a particular function key is entered and it cannot fit into the space used by the old definition the whole buffer is sorted, the other definitions moved up and the new one tacked on the end. If the new definition can fit into the old space then it is put there and the other definitions left as they are. Sometimes this process can result in some wasted space in the buffer meaning that you get the error message 'Bad key' (signifying that there is no room for your definition) when you enter a new definition even though there is actually room for it.

ABBREVIATIONS

With knowledge of the buffer arrangement it can be appreciated that in order to get the maximum number of keys defined we must use abbreviations and control codes wherever possible.

A list of Basic keyword abbreviations was printed in ELBUG Vol.1 No.7, but each can also be found in the User Guide alongside its Basic keyword.

```
*KEY10 OLD|M
*KEY10 O.|M
```

This function key definition will redefine the Break key automatically to restore a Basic program if Break is pressed. The second definition uses the abbreviation for OLD (O.) and saves a byte by doing so.

```
*KEY1 |V6|NL.|M
```

This definition does the same job as the key 1 definition earlier in the article. Now only 6 bytes of the function key buffer are used instead of 12 when the fully expanded version was used - a saving of 50%. (|V = Ctrl-V = MODE and L.= LIST).

The error line listing definition detailed before occupies 90 bytes. This can be reduced to 76 by removing all spaces and by using abbreviations.

However, note that not all Basic keyword abbreviations save space. For example, CHR$ and CHR. both use 4 bytes.

Next month we shall continue our look at the Electron's function keys with methods to make your definitions invisible, to save your function key definitions on cassette and other useful hints.

Your Guide To BEEBUG

None of the programs appropriate to the Electron in this month's BEEBUG need any alteration to run. However, a few general comments on Beeb program conversion may prove useful to all Electron users.

Beeb programs sometimes fail to run properly on the Electron because of the Beeb's mode 7 and extra sound facilities. However, conversion rarely proves very difficult.

The Beeb's mode 7 display is of the same format as the mode 6 display that ELBUG members know but with a full range of colours and crude graphics. Mode 7 is not available on the Electron. However, the Electron defaults to mode 6 when it encounters a MODE 7 command. So BBC micro programs that include a mode 7 display will therefore run on the Electron but often with a corrupted (and monochrome) display.

The Beeb has three sound channels each with full control over the volume of notes. Electron Basic includes all the SOUND and ENVELOPE parameters that are used in the BBC micro even though several have no effect. This means that Beeb programs making full use of sound will still run on the Electron unaltered, though not, of course, producing the full sound effect.

The other major difference between the Electron and the BBC micro is speed. The Electron is much slower, and particularly slow in modes 0, 1, and 2. BBC micro programs often benefit from a change of mode (admittedly, with a loss of some colours) when running them on the Electron. When changes are needed they will be given in this ELBUG supplement.

ACORNSOFT'S GRAPHICS EXTENSION ROM

The graphics extension ROM from Acornsoft does not operate correctly on the Electron. However, the chances are that Acorn will soon release a version for the Elk on a cartridge suitable for the Plus-1. Meanwhile, Computer Concepts' graphics extension ROM does operate on the Elk and it provides many of the facilities noted here.

POST-HASTE

ELBUG members who are lucky enough to own an RS423 cartridge modem from Pace will need no encouraging to delve into the delights of electronic mail. However, they may find this account of the varied services available enlightening.

FULL-SCREEN BASIC PROGRAM EDITOR

Although the Electron has a line editor built-in to its Basic, this does not provide the easiest way of editing a Basic program. This extremely short but very useful program will make writing programs on the Electron so much easier.

The editor operates on the Electron with no changes to the published listing.

BEEBUG FILER (Part 2)

This second article continues the construction of this most useful database management program.

However, the program requires the facility of random access to files. This is something a cassette based system (like the unadorned Electron) cannot provide so only ELBUG members lucky enough to own a Plus-3 (or any other) disc system will be able to use this program on their Electron.

UPGRADING YOUR BEEB

Acorn 1770 and ADFS

Unlike the BBC micro, the Electron has no need for these upgrades. The Plus-3 already provides the Electron with a 1770 disc controller chip and an ADFS, as standard.

PERSONAL DIARY MANAGER

This useful diary program will help you to keep all those important dates in the coming year. The first date you could make is to go out and buy a Plus-3 disc interface as this program requires random access of discs to operate.

If you have a disc interface connected to your Electron, the program will run. It makes extensive use of mode 7 so the display will be a little corrupted. However, this will only affect the colour

and the (purely decorative) graphics.

SURE SAVE UTILITY

This handy utility can only be made to operate with a machine equipped with a disc drive and running the BBC micro's DFS. No luck for Electron users here.

GAMES REVIEWS

Two of the games reviewed here are available in Electron versions. These are the excellent DeathStar and Confuzion.

FIRST COURSE - Colour Blending

This useful technique for squeezing even better graphics out of an Electron will appeal to most ELBUG members. Although the Elk is restricted to displaying only eight colours by surreptitious mixing of these, many shades can be created. This article, and the demonstration program, will show you how it's all done.

PROGRAMMING WITH WORDWISE PLUS (Part 3)

Unfortunately the Wordwise Plus word processor is not available for the Electron so this excellent tutorial series will not be of interest to ELBUG members.

WORKSHOP - Sorting Revisited

Those members who read the three Workshop articles on sorting and searching, a few issues back (BEEBUG Vol.3 No.10 to Vol.4 No.2) will be interested to see this follow-up article.

The demonstration program presented here will not run on the Electron quite as it does on the Beeb, as it directly accesses a mode 7 screen. However, the sorting procedures are quite applicable to the Electron and the timings that the program produces will show you the advantages of a correctly chosen sorting method, as intended.

PROGRAMMING SIDEWAYS RAM AND ROM

Although the concept of sideways software, in either ROM or RAM, is equally applicable to the Electron as to the BBC micro, this utility program to create sideways software will not run on the Electron because it requires the memory economy of the Beeb's mode 7 display.

BOOK REVIEWS

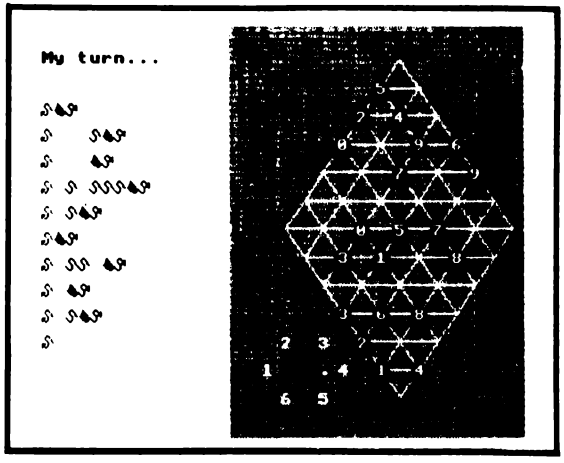
Several of the books reviewed this month could rightfully grace the shelves of any Electron owner.

The BBC Micro and Electron Book is an excellent guide to both machines and is recommended for ELBUG members.

Within the BBC Micro is a bit too specific to the Beeb for Electron owners but is well worth a flick through at the library.

The 6502 Reference Guide is, of course, as applicable to the Electron as it is to any other computer with this processor.

The Hitch-Hiker's Guide to Artificial Intelligence is a useful introduction to this subject that will appeal to Electron owners as much as their Beeb counterparts.



CHINESE CHEQUERS

This entertaining version of the old board game will work perfectly on the Electron and keep as many ELBUG members as BEEBUG members awake after the plum pudding.

HINTS

The following hints from this month's BEEBUG are applicable to ELBUG members:

MORE USER DEFINED CHARACTERS
TESTING OSBYTE ROUTINES
STIPPLED LINES
CLEAR DESCRIPTION

STARWORD AND STARSTORE FROM SLOGGER

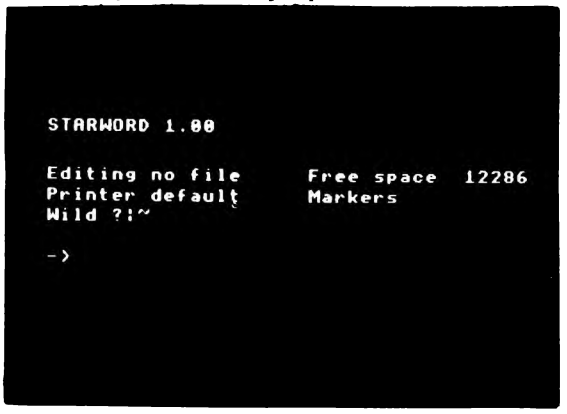
Alan Webster takes a look at the latest ROMs from Slogger - a wordprocessor and a database package.

Product : Starword
Supplier: Slogger Ltd,
215 Beacon Road,
Chatham, Kent.
(0634) 811634
Prices : £34.50

Starword is a ROM based wordprocessor very similar in use to Acornsoft's View. It is supplied on a 16k EPROM along with a comprehensive and well written manual and can be used on any Electron fitted with a suitable ROM board or ROM adaptor cartridge.

The wordprocessor will allow you to create, edit and format documents, and save them to cassette or disc. There are two parts to Starword: the system mode and the text mode.

Upon typing *STARWORD (or *WORD) from any language you will enter Starword's system mode. This presents you with the following screen display:

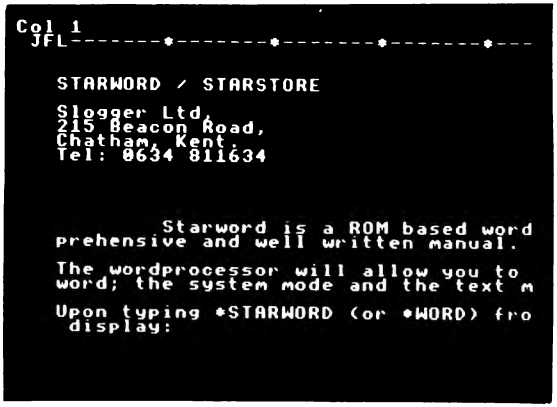


This shows that no filename has been given for loading and saving, that there are 12286 bytes free and the default Epson printer driver is active. If you own a different type of printer, a printer driver generator is available for you to produce your own driver. This costs £7.95 on cassette or £11.95 on disc.

From system mode you are allowed to type any operating system (*) commands (i.e. to return to Basic or get a catalogue). The screen display defaults to mode 3 (80 column text), but can be changed to mode 6 by the command '40' and back to mode 3 by typing '80'. Text is shown in green, but this can also be changed by using the command 'COLOUR'.

The text mode is the main part of the wordprocessor as this allows you to enter and edit your text. Text mode is entered by pressing Escape whilst in system mode (likewise, pressing Escape from text mode will return you to system mode).

Once in text mode you will see a display similar to the figure below. The column number displays the cursor position within the main text window. The status line will tell you whether you are in insert or overwrite mode, whether justification is on or off and whether formatting is on or off. The ruler display will show the current ruler size with all Tab positions and left and right margins. The stored commands window is specifically for text formatting purposes and holds commands to tell Starword how to format the text. Some of these commands are described later.



The message window at the bottom of the screen displays all messages and prompts without overwriting your main text which is entered into the main text window.

Once in text mode, you can start to type in your text. When you reach the edge of the ruler (the right hand margin) the text will automatically wrap onto the next line of the screen without splitting the word. If there were only three spaces left on the line, the word 'at' would fit, but the word 'computer' would begin on the next line. If justification is on, then once the text starts a new line, the previous one will be padded with spaces to line up the right hand edge with the right hand margin.

If you have finished typing your piece of text, but notice a spelling mistake, then you just use the cursor keys to move to the appropriate word, and you can simply correct the spelling. The two modes, overwrite and insert, tell you how the text will be entered. If in insert mode, any new text will be inserted at the cursor and the characters to the right shifted up accordingly. In overwrite, the text will overwrite the character at the cursor's position.

Blocks of text can be selected with markers and then deleted, copied, or moved around the document making editing text very easy. In addition, Starword can perform a search and replace operation through the document to change all occurrences of one string to another.

The manual is clearly written and straightforward to follow. Practical examples are given and you are taken through every aspect of the wordprocessor at a gentle pace. One good thing about the manual is that it doesn't force you to read everything before you can start creating documents easily. The first quarter of the manual is a tutorial to get you started and the rest is taken up with a description of the various commands.

Before creating a document you should set up the text ruler to give you a set line length and margins. The text ruler can be altered at any time (in both system and text modes). In system mode, the left and right margins are altered with commands, whilst in text mode they can be changed by the use of function key 1.

Other commands restore the default ruler, and clear all the tab stops or set them to default positions.

The wordprocessor has on-screen formatting, which means what you see on the screen is just what will appear on the final printout.

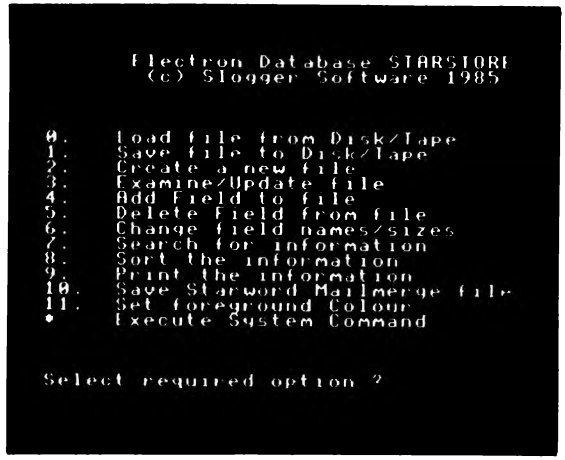
Starword is much easier to use than Acornsoft's View (especially for a beginner to wordprocessing), but doesn't quite have the same professional feel to it once you have been using it for a while. If you are thinking of buying a wordprocessor for your Electron, Starword is certainly worth considering. If you have only just started wordprocessing then this would be an ideal choice.

Product : Starstore
Supplier: Slogger Ltd,
215 Beacon Road,
Chatham, Kent.
(0634) 811634
Prices : £29.95

Starstore is a ROM based database system which will allow Electron owners to store, retrieve, analyse, search, sort and print large quantities of data very quickly.

The package allows you to create a database with a maximum of 255 records of up to 30 fields with up to 127 characters per field. A database could be used, for example, as an address book, or to hold details of a record or tape collection.

Starstore is invoked through the command *STORE and will present you with a menu page as in the figure below.


Electron Database STARSTORE
(c) Slogger Software 1985
0. Load file from Disk/Tape
1. Save file to Disk/Tape
2. Create a new file
3. Examine/Update file
4. Add Field to file
5. Delete Field from file
6. Change field names/sizes
7. Search for information
8. Sort the information
9. Print the information
10. Save Starword Mailmerge file
11. Set foreground Colour
* Execute System Command
Select required option ?

One of the nice things about this database is the extensive error trapping. This means that in most instances the user cannot get into any trouble when using the system.

Starstore works with both cassette and disc systems, but only works with the data currently in memory (a mode 6 display is always used to maximize the remaining memory). This means that disc users get no extra benefit from the fact that the DFS supports random access files.

One small omission that I noticed very early was that there was no free memory display, and this became annoying when I got quite near to the end of my record collection, and suddenly when trying to add more fields to my file I ran out of memory. Tape users have a total of 20736 characters spare at the beginning and disc users have 16896 characters spare.

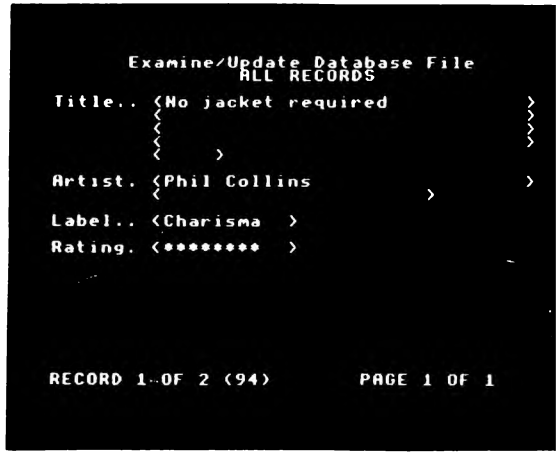
Starstore has a useful mailmerge facility, enabling records from the database to be output to a file for later use in Starword. You can, for example, type in a single letter in Starword and then using a file of names and addresses, printout individual letters correctly headed and addressed.

To enter a database file, you must first use option 2 to create a file. This will prompt you for a field name and field size for up to 30 fields. The field size can be up to 127 characters, but the field name can only consist of 7 characters, so an address book cannot have a field name of 'post code'. To set up a file of music albums, a typical file would have the following field names and sizes:

Field Name	Field Size
Title	100
Artist	40
Label	10
Rating	10

To exit the set up procedure before 30 fields have been defined, simply press Escape.

Once the file has been set up, you can now go on to examining and updating this file using menu option 3. Then we can begin to enter a record collection. Using the cursor keys to move around within a record, you can enter your data for a particular album or single.



When you have finished entering data into the first record, you can move the cursor down to the last field and press Return to go on to the next record. Pressing Escape at any time will return you to the main menu.

After you have finished entering your records you can save them to tape or disc, search them, sort them or print them.

Overall Starstore is a standard database, with some nice features, and has the advantage of being compatible with Starword. If you want both a database and a wordprocessor for your Electron, then I recommend these. ●